



Introduction to Theory of Computation

CMPS 3140 LECTURE 1



Theory of Computation

- Branch of computer science
 - “What are the fundamental capabilities and limitations of a computer?”
 - Studies **how efficiently** a problem can be solved using a **model of computation**
 - Model in which describes how a **set of outputs are computed given a set of inputs**.
 - Using an algorithm, to what degree can a problem be solved?



Branches of Computation

- Divided into 3 major areas:
 1. Automata
 2. Computability
 3. Complexity
- What are the fundamental capabilities and limitations of computers?
 - Each area interprets this question differently
 - Answer varies with each



Automata Theory

- **Automatons/Automata**

- Means “something that is doing something by itself”
- Abstract models of machines that perform computations on an input by moving through a series of states or configurations

- **Objective of Automata Theory**

- Describing abstract models can be difficult
- Develop methods by which computer scientists can describe and analyze the dynamic behavior of discrete systems.
 - Discrete systems that can take distinct states



Computability Theory

- Some problems can not be solved by computers
 - Examples
 - Determining whether any given mathematical statement is either true or false.

- Computability Theory
 - Studies whether a problem is solvable (computable) or not
 - Is there a possible algorithm to solve this problem?

 - To what extent is a problem solvable
 - Can we reduce the problem to solve it?



Complexity Theory

- Different problems have different complexities
 - Sorting problem
 - Single simple criteria
 - Scheduling problem
 - Multiple constraints for each individual schedule
- Complexity Theory
 - Classify problems as easy or hard to solve
 - What makes some problems computationally hard and others easy?
 - No single answer
 - Unnecessary to prove that a problem is hard
 - Can give evidence that a problem is hard



Complexity of a Problem

- Problem types

- Easier solutions are usually preferable
 - Problems that have solutions but that are too complex may not be viable
- Alternatively, fields like Cryptography prefer harder solutions for their needs

- Options for hard problems

- Alter the problem to be more easily solvable
- Consider alternate types of solutions or computations
- Find a suboptimal solution
- Don't optimize for worst case situation of problem
 - Solution may be fast in most cases except for specific scenarios.